Remarks

This Application has been carefully reviewed in light of the Final Office Action mailed March 30, 2004, and the Advisory Action mailed September 23, 2004. Applicants appreciate the Examiner's consideration of Applicants' previous Response. Although Applicants believe all pending claims are allowable without amendment, Applicants have made clarifying amendments to Claims 1-2, 4-5, 7-10, 13-15, 17, 20-23, and 25. Certain of these amendments are not considered narrowing, and none are considered necessary for patentability. Applicants respectfully request reconsideration and allowance of all pending claims.

I.   Claims 7-9 are Allowable over *Lection*

The Examiner rejects Claims 7-9 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent 6,418,446 to Lection, et al. (*"Lection"*). Applicants respectfully disagree.

Lection discloses "a method, system, and computer-readable code for grouping dynamic schema data using Extensible Markup Language notation." (Column 1, Lines 6-10) More specifically, *Lection* discloses a method "to gather data that may have had changes to its format, and create a structured representation of this data that flexibly adapts to format variations" and that "a DOM tree created from an XML representation of the source data is used by the present invention as it creates an output DOM tree." (Abstract and Column 1, Lines 6-10) The technique disclosed in *Lection* includes providing an input data source comprising one or more records each having this dynamically variable record format, and wherein the dynamically variable record format of each record comprises a plurality of dynamically variable fields; processing a gather verb specification that identifies a selected one of the data records (which may be formatted as a first DOM tree) from the input data source and an output data destination; gathering the dynamically variable fields from the selected one of the records according to the gather verb specification (which may be formatted as a second DOM tree); and transferring the gathered dynamically variable fields to the output data destination (which may be formatted as a third DOM tree) according to the gather verb specification. (Column 3, Lines 34-46 and 48-52) The first and second DOM

trees may be created by parsing an XML representation of the selected record and the gather verb specification, respectively. (Column 3, Lines 52-56)

In contrast, independent Claim 7, for example, recites:

> A method for outputting data as Extensible Markup Language from an application running on a computer system, the method comprising:
>
> establishing a relationship of the output data and one or more Extensible Markup Language Document Object Model contexts, the output data corresponding to a write operation of the application for outputting the data, the one or more Extensible Markup Language Document Object Model contexts identifying a position in a target Extensible Markup Language schema of the output data corresponding to the write operation of the application;
>
> building a Document Object Model instance with the one or more Extensible Markup Language Document Object Model contexts; and
>
> outputting the data from the Document Object Model instance as Extensible Markup Language.

Applicants respectfully submit that *Lection* fails to disclose, teach, or suggest various aspects of independent Claim 7.

At the outset, Applicants note that *Lection* does not even relate to "outputting data as Extensible Markup Language from an application running on a computer system," as recited in Claim 7. Instead, *Lection* is directed to a process, system, and method for gathering data having dynamically variable record formats such as those created when a dynamic schema is used with a data repository. (Column 3, Lines 30-34) In other words, *Lection* is directed to mapping stored data records, which are already structured in XML format, to variable record formats. The Examiner appears to equate "the source data" of *Lection*, which is stored as a structure data record, with "the output data" in Claim 7. (*See* Office Action, Page 3) Applicants respectfully submit that *Lection* does not support this interpretation. The source data disclosed in *Lection* is merely a stored structured data record, it is not data output from an application running on a computer system, and it does not correspond to a write operation of the application as recited in Claim 7. Thus, *Lection* fails to disclose, teach, or suggest "establishing a relationship of the output data and one or more Extensible Markup Language Document Object Model contexts, the output data corresponding to a write operation of the

application for outputting the data, the one or more Extensible Markup Language Document Object Model contexts identifying a position in a target Extensible Markup Language schema of the output data corresponding to the write operation of the application," as recited in Claim 7 as amended.

As disclosing this limitation prior to the amendments presented in this Response, the Examiner cited the following statement from *Lection*: It is another object of the present invention [the invention disclosed in *Lection*] to provide this technique using a DOM tree created from an XML syntax representation of the source data. (*See* Office Action, Page 3; Column 3, Lines 8-10) The Examiner then states, "DOM tree is the relationship of the output data." (Office Action, Page 3) Applicants respectfully disagree and submit that the DOM tree disclosed in *Lection* is a representation of an XML representation of the source data, a stored structured data record, not the output data of an application that corresponds to a write operation of the application as recited in Claim 7, as amended. The DOM tree disclosed in *Lection* provides the relationship of items in the stored data record, which may already be in XML. In *Lection*, there is no establishment of a relationship of the output data [the output data corresponding to a write operation of an application for outputting the data] and one or more Extensible Markup Language Document Object Model contexts "identifying a position in a target Extensible Markup Language schema of the output data," as recited in Claim 7 as amended.

At least because *Lection* fails to disclose, teach, or suggest the output data as recited in Claim 7, *Lection* necessarily fails to disclose, teach, or suggest "establishing a relationship of the output data and one or more Extensible Markup Language Document Object Model contexts" and "building a Document Object Model instance with the one or more Extensible Markup Language Document Object Model contexts," as recited in Claim 7 as amended. The Examiner seems to equate "an output DOM tree" of *Lection* with "a Document Object Model instance" recited in Claim 7. The Examiner further appears to equate "the source data" of *Lection* with "the output data" in Claim 7. (*See* Office Action, Page 3)

In response to Applicants' arguments presented in the previous two Responses, the Examiner stated that the DOM tree is the relationship of the output data and that the "Examiner interprets the representation of the selected record and/or representation of the gather verb specification is context." (*See* Office Action, Pages 3 and 31) Applicants respectfully submit that *Lection* does not support this interpretation. The first DOM tree disclosed in *Lection* represents the structured source data record, which, as Applicants discussed above, cannot be properly equated with the "output data" recited in Claim 7. The GATHER verb specification disclosed in *Lection* and cited by the Examiner is an API formatted as a DOM tree that merely provides a way "to gather data that may have had changes to its format, and create a structured representation of this data that flexibly adapts to format variations." (*See* Abstract; Column 3, Lines 49-51; Column 10, Line 66 through Column 11, Line 2; and Column 18, Lines 49-52) The XML representation of the selected record and the XML representation of the GATHER verb specification disclosed in *Lection* are merely XML representations of structured data and do not disclose, teach, or suggest contexts as recited in Claim 7.

Applicants respectfully note that "[a] claim is anticipated only if *each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987) (emphasis added); M.P.E.P. § 2131. Stated another way, "for anticipation under 35 U.S.C. 102, the reference must teach *every aspect* of the claimed invention either explicitly or impliedly." M.P.E.P. § 706.02 (emphasis added). In addition, "*[t]he elements must be arranged as required by the claim.*" M.P.E.P. § 2131 (emphasis added) referencing *In re Bond*, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990); *see also Richardson v. Suzuki Motor Co.*, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989). Furthermore, "[t]he *identical invention* must be shown in as complete detail as is contained in the . . . claim." M.P.E.P. § 2131 citing *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236 (Fed. Cir. 1989) (emphasis added). As illustrated above, *Lection* fails to disclose, either expressly or inherently, each and every limitation recited in Claim 7, as is required under the M.P.E.P. and governing Federal Circuit cases.

For at least these reasons, *Lection* fails to disclose, teach, or suggest various limitations of independent Claim 7, as amended. Accordingly, Applicants respectfully request reconsideration and allowance of independent Claim 7 and its dependent claims.

II.   **Claims 23 and 25 are Allowable over *Stefaniak***

The Examiner rejects Claims 23-25 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent 6,550,054 to Stefaniak, et al. (*"Stefaniak"*). Applicants respectfully disagree.

Claim 23, for example, recites:

> A method for modeling a legacy computer system, comprising:
> identifying incidents within one or more applications of the legacy computer system that output data;
> associating the identified output incidents within the one or more applications with an Extensible Markup Language schema;
> defining a control flow graph of the identified output incidents;
> based at least on the association of the identified output incidents within the one or more applications with the Extensible Markup Language schema, creating a modification specification for modifying the one or more legacy computer system applications to provide output from a Document Object Model instance as Extensible Markup Language; and
> automatically modifying, based at least on the modification specification, the one or more legacy computer system applications such that the one or more modified legacy computer system applications provide output from the Document Object Model instance as Extensible Markup Language.

Applicants respectfully submit that *Stefaniak* fails to disclose, teach, or suggest various limitations recited in independent Claim 23.

For example, *Stefaniak* fails to disclose, teach, or suggest "identifying incidents within one or more applications of the legacy computer system that output data," as recited in Claim 23 as amended. Applicants made a similar argument in the previous Response, but it appears that the Examiner did not consider this argument, as the Examiner did not respond to this argument in the Advisory Action. Applicants reiterate that, at best, *Stefaniak* merely discloses capturing and recording screen relationships after an application has generated the screen output. (*See* Column 1, Lines 41-47; Column 3, Lines 60-65; Column 5, Lines 41-45)

Thus, it appears that the system disclosed in *Stefaniak* is merely aware that screen output has been generated after it has been generated. In fact, it appears that the system disclosed in *Stefaniak* remains outside the programs and merely captures screen output. Nowhere does *Stefaniak* disclose, teach, or suggest that it "identif[ies] incidents within one or more applications of the legacy computer systems that output data," as recited in Claim 23 as amended.

As another example, *Stefaniak* fails to disclose, teach, or suggest "defining a control flow graph of the identified output incidents," as recited in Claim 23 as amended. As disclosing this limitation (prior to the amendments presented in the current Response), the Examiner cites to Figure 3 of *Stefaniak* and its associated text. (*See* Office Action, Page 5) First, at least because *Stefaniak* fails to disclose, teach, or suggest "identifying incidents within one or more applications of the legacy computer system that output data," as recited in Claim 23, *Stefaniak* necessarily fails to disclose, teach, or suggest "defining a control flow graph of the identified output incidents," as recited in Claim 23. Second, Figure 3 of *Stefaniak* has nothing to do with "defining a control flow graph," as recited in Claim 23. Instead, Figure 3 of *Stefaniak* and its associated text merely illustrate an end-to-end process flow from a legacy program to a UML model. (*See* Column 2, Lines 29-30 and Column 5, Lines 39-41) In other words, the cited portion of *Stefaniak* merely shows a process of how to generate a UML model of a legacy program according to the system disclosed in *Stefaniak*. If the Examiner maintains this argument, Applicants respectfully request that the Examiner provide an explanation for how this disclosure of *Stefaniak*, which has nothing to do with generating a control flow graph, discloses, teaches, or suggests "defining a control flow graph of the identified output incidents," as recited in Claim 23 as amended.

As another example, *Stefaniak* fails to disclose, teach, or suggest "associating the identified output incidents within the one or more applications with an Extensible Markup Language schema," as recited in Claim 23 as amended. First, at least because *Stefaniak* fails to disclose, teach, or suggest "identifying incidents within one or more applications of the legacy computer system that output data," as recited in Claim 23, *Stefaniak* necessarily fails to disclose, teach, or suggest "associating the identified output incidents within the one or

more applications with an Extensible Markup Language schema," as recited in Claim 23 as amended. Second, *Stefaniak* merely discloses creating an XML representation of *the UML model* of a terminal application. (*See* Figure 9 and Column 8, Lines 28-47) Nowhere does the cited portion of *Stefaniak* disclose, teach, or suggest "associating the identified output incidents within the one or more program applications with an Extensible Markup Language schema," as recited in Claim 23 as amended.

As another example, *Stefaniak* fails to disclose, teach, or suggest "based at least on the association of the identified output incidents within the one or more applications with the Extensible Markup Language schema, creating a modification specification for modifying the one or more legacy computer system applications to provide output from a Document Object Model instance as Extensible Markup Language," as recited in Claim 23 as amended. First, at least because *Stefaniak* fails to disclose, teach, or suggest "identifying incidents within one or more applications of the legacy computer system that output data," as recited in Claim 23, *Stefaniak* necessarily fails to disclose, teach, or suggest this limitation. Second, because *Stefaniak* does not disclose, teach, or suggest "associating the identified output incidents within the one or more program applications with an Extensible Markup Language schema," as recited in Claim 23 as amended, *Stefaniak* necessarily fails to disclose, teach, or suggest this limitation. Third, *Stefaniak* does not even discuss Document Object Models; thus, *Stefaniak* clearly fails to disclose, teach, or suggest "creating a modification specification for modifying the one or more legacy computer system applications to provide output from a Document Object Model instance as Extensible Markup Language," as recited in Claim 23 as amended.

Moreover, the portion of *Stefaniak* on which the Examiner relies to reject this limitation (prior to the amendments presented in this Response), which Applicants note merely includes brief descriptions of two drawings, states, "FIG. 6 is a flow chart depicting the process of generating an XML file representation of a UML model created by the process described in the process shown in FIGS. 5A through 5C. FIGS. 7A and 7B combined form a flow chart depicting the process of parsing a text file to generate an object-oriented representation of the UML model in memory." (Column 2, Lines 38-44; *see* Office Action,

Page 5) The cited portions apparently disclose creating an XML representation of a UML model (which is a text file) of an application. This disclosure clearly fails to disclose, teach, or suggest "based at least on the association of the identified output incidents within the one or more applications with the Extensible Markup Language schema, creating a modification specification for modifying the one or more legacy computer system applications to provide output from a Document Object Model instance as Extensible Markup Language," as recited in Claim 23 as amended.

As another example, Applicants maintain that *Stefaniak* fails to disclose, teach, or suggest "automatically modifying the legacy computer system applications in accordance with the specification," as recited in Claim 23 prior to the amendments presented in the present Response, and Applicants reiterate their arguments presented in previous Responses pertaining to these distinctions. For example, Applicants maintain that *Stefaniak* fails to support the Examiner's assertion that "transforming a terminal-based screen application into an application specification" in *Stefaniak* can be equated with "automatically modifying the legacy computer system applications in accordance with the specification" as recited, in part, in amended Claim 23. (*See* Office Action, Pages 5-6 and 31) *Stefaniak* discloses a system that describes legacy application screens in terms of a terminal application specification and <u>converts</u> the specification into a UML <u>model</u>. (*See* Abstract and Column 1, Lines 57-67) *Stefaniak* repeatedly teaches that the output of the system is a <u>representation</u> or <u>model</u> of the terminal-based application – there is no modification of the terminal-based application in *Stefaniak*. (*See* Title; Abstract; Column 1, Lines 15-18 and 28-31) This is further suggested by *Stefaniak*'s continued use of UML, a modeling language, for <u>representing</u> or <u>modeling</u> – as opposed to modifying – the terminal-based application.[1] In other words, even if "the terminal-based application" in *Stefaniak* is comparable to "the legacy computer system applications" of Claim 23 (which Applicants do not concede), *Stefaniak* fails to disclose, teach, or suggest "automatically modifying the legacy computer system applications in accordance with the specification" as recited, in part, in amended Claim 23.

---

[1] For example, the "Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems." UML specification, available at <u>www.omg.com/uml</u>.

In the Advisory Action, the Examiner referenced certain disclosures in *Stefaniak* and then provided a purported example of those disclosures -- an example that is not provided in *Stefaniak*. Applicants respectfully submit that the example provided by the Examiner is not supported by the teachings of *Stefaniak*. In particular, the Examiner states, "For example, application I contains core program module and subroutine I for display of output data in a text-based environment, and application II contains core program and subroutine II for displaying output data in a GUI environment. Since subroutine I is different than subroutine II . . . application program I is not equal to application program II as a whole." (Advisory Action, Continuation Sheet) Applicants do not agree that there is any such application I and application II disclosed in *Stefaniak*, where application I produces output in text format and application II produces output in a format for a GUI. Instead, *Stefaniak* discloses "transforming a terminal-based screen application into an application specification," "converting the application specification into a modeling language-based representation" (i.e., the UML model), and "displaying the modeling language based representation [i.e., the UML model] with a graphical user interface." (Abstract) Their simply is no "application II," as stated by the Examiner. In *Stefaniak*, an application is transformed into an application specification, which is then converted in a UML model, which is then displayed with a GUI.

Moreover, Applicants respectfully submit that *Stefaniak* is merely generating models of legacy application screens - it is not modifying legacy computer applications. Whatever words are used in *Stefaniak*, the context makes it clear that the system disclosed in *Stefaniak* is merely generating a model of the legacy application screens; it does not appear to be modifying applications of the legacy computer system. Second, the first sentence cited by the Examiner states converting the specification into a UML model, not the legacy applications. Third, *Stefaniak* discloses the following method: (1) transforming a terminal-based application into an application specification; (2) converting the *application specification* into a modeling language-based representation (e.g., UML); and (3) displaying the modeling language-based representation with a graphical user interface. (*See* Abstract) The Examiner argues that "transforming a terminal based screen application into an application specification" discloses "automatically modifying the legacy computer system applications in

accordance with the specification." (Office Action, Page 5) Assuming for the sake of argument only that this equation is even possible (which it is not as Applicants discussed above), how could the modification be "in accordance with the specification," as recited in Claim 23, when the modification is the creation of the specification (as argued by the Examiner)? Applicants respectfully submit that it could not. Alternatively, assuming for the sake of argument only that the application specification disclosed in *Stefaniak* could be equated with the modification specification recited in Applicants' Claim 23 (which Applicants do not concede, particularly in light of the fact that Applicants' Claim 23 recites "creating a modification specification for modifying the legacy computer system applications"), how could *Stefaniak* disclose, teach, or suggest "automatically modifying, based at least on the modification specification, the one or more legacy computer system applications in accordance with the specification such that the one or more modified legacy computer system applications provide output from the Document Object Model instance as Extensible Markup Language" if the Examiner is equating the creation of the application specification disclosed in *Stefaniak* with the automatic modification of the legacy computer system applications recited in Applicants' Claim 23? Applicants respectfully submit that it cannot.

In any event, *Stefaniak* certainly fails to disclose, teach, or suggest "automatically modifying, based at least on the modification specification, the one or more legacy computer system applications such that the one or more modified legacy computer system applications provide output from the Document Object Model instance as Extensible Markup Language," as recited in Claim 23 as amended. There is no modification of any application in *Stefaniak*, and even more clearly, there is no automatic modification of one or more legacy computer system applications based at least on the modification specification as recited in Claim 23. Furthermore, *Stefaniak* does not even discusses a Document Object Model. Thus, *Stefaniak* necessarily fails to disclose, teach, or suggest "automatically modifying . . . the one or more legacy computer system applications such that the one or more modified legacy computer system applications provide output from the Document Object Model instance as Extensible Markup Language," as recited in Claim 23 as amended. Again, Applicants made a substantially similar argument in the previous Response, but it appears that the Examiner did

not consider this argument, as the Examiner did not provide a response to this argument in the Advisory Action.

Applicants reiterate the legal standard for a finding of anticipation discussed above with reference to independent Claim 7. As illustrated above, *Stefaniak* fails to disclose, either expressly or inherently, each and every limitation recited in Claim 23, as is required under the M.P.E.P. and governing Federal Circuit cases.

For at least these reasons, *Stefaniak* fails to disclose, teach, or suggest various limitations of independent Claim 23. Moreover, independent Claim 25 is allowable at least for analogous reasons. Accordingly, Applicants respectfully request reconsideration and allowance of independent Claims 23 and 25.

III. **The Claims are Allowable over the Various Rejections under 35 U.S.C. § 103**

The Examiner rejects:

- Claims 10-14 under 35 U.S.C. § 103(a) as being unpatentable over *Lection* in view of *Stefaniak*;

- Claims 15-18 under 35 U.S.C. § 103(a) as being unpatentable over *Lection*, in view of *Stefaniak*, further in view of Shanmugasundaram et al., "Relational Databases for Querying XML Documents: Limitations and Opportunities" ("*Shanmugasundaram*"); and

- Claim 19 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Lection* in view of *Stefaniak*, further in view of *Shanmugasundaram*, further in view of U.S. Patent No. 6,209,124 to Vermeire et al ("*Vermeire*").

Applicants respectfully traverse these objections and all assertions and holdings therein. For at least the reasons discussed above with respect to Claim 7, *Lection* fails to disclose, teach, or suggest various aspects of Claims 10-14 and 15-19. Further, *Stefaniak*, *Vermeire*, and/or *Shanmugasundaram*, whether considered individually or in combination, fail to account for the deficiencies of *Lection*. Furthermore, Claims 10-19 recite further patentable distinctions over the various combinations of references proposed by the

Examiner. To avoid burdening the record and in view of the clear deficiencies of *Lection*, Applicants do not specifically discuss these distinctions in this Response. However, Applicants reserve the right to discuss these distinctions in a future Response or on Appeal, if appropriate. Accordingly, Applicants respectfully request reconsideration and allowance of Claims 10-19.

The Examiner further rejects:

- Claims 1-2, 4-6, 20-21 under 35 U.S.C. § 103(a) as being unpatentable over *Stefaniak*, in view of U.S. Patent No. 6,618,852 to van Elkeren et al ("*van Elkeren*");

- Claim 3 under 35 U.S.C. § 103(a) as being unpatentable over *Stefaniak* in view of *van Elkeren*, further in view of U.S. Patent No. 6,347,307 to Sandhu et al ("*Sandhu*"); and

- Claim 22 under 35 U.S.C. § 103(a) as being unpatentable over *Stefaniak* in view of *van Elkeren*, further in view of *Shanmugasundaram*.

Applicants respectfully traverse these objections and all assertions and holdings therein. For at least the reasons discussed above with respect to Claim 23, *Stefaniak* fails to disclose, teach, or suggest various aspects of Claims 1-6 and 20-22. Further, *van Elkeren*, *Sandhu*, and/or *Shanmugasundaram*, whether considered individually or in combination, fail to account for the deficiencies of *Stefaniak*. Furthermore, Claims 1-6 and 20-22 recite further patentable distinctions over the various combinations of references proposed by the Examiner. To avoid burdening the record and in view of the clear deficiencies of *Stefaniak*, Applicants do not specifically discuss these distinctions in this Response. However, Applicants reserve the right to discuss these distinctions in a future Response or on Appeal, if appropriate. Accordingly, Applicants respectfully request reconsideration and allowance of Claims 1-6 and 20-22.

Furthermore, with respect to all of the proposed combinations of references made by the Examiner, Applicants do not admit that the proposed combinations of references are possible or that the Examiner has demonstrated the required teaching, suggestion, or

motivation to combine these references. For at least these reasons, Applicants respectfully request reconsideration and allowance of Claims 1-6, 10-19, and 20-22.

III. <u>No Waiver</u>

All of Applicants' arguments and amendments are without prejudice or disclaimer. Additionally, Applicants have merely discussed example distinctions from the various references cited by the Examiner. Other distinctions may exist, and Applicants reserve the right to discuss these additional distinctions in a later Response or on Appeal, if appropriate. By not responding to additional statements made by the Examiner, Applicants do not acquiesce to the Examiner's additional statements. The example distinctions discussed by Applicants are sufficient to overcome the Examiner's rejections.
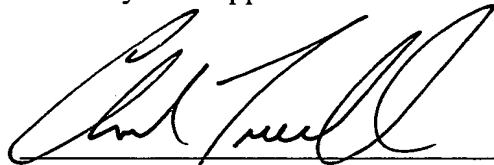
Conclusion

Applicants have now made an earnest attempt to place this case in condition for immediate allowance. For the foregoing reasons and for other apparent reasons, Applicants respectfully request allowance of all pending claims.

If the Examiner feels that prosecution of the present Application may be advanced in any way by a telephone conference, the Examiner is invited to contact the undersigned attorney at 214.953.6813.

Applicants enclose a check in the amount of $950.00 to cover the cost of a three-month extension of time. The Commissioner is hereby authorized to charge any deficiency or credit any overpayment to Deposit Account No. 05-0765 of Electronic Data Systems Corporation.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Applicants

Chad D. Terrell
Reg. No. 52,279

Date: September 30, 2004

Correspondence Address:

2001 Ross Avenue
Dallas, TX 75201-2980
Telephone No. 214.953.6813

Customer Number: **35005**